RTC
应用指南

文档版本　10

发布日期　2019-05-15

商标声明

 、HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址：　深圳市龙岗区坂田华为总部办公楼　邮编：518129

网址：　http://www.hisilicon.com/cn/

客户服务邮箱：　support@hisilicon.com

前 言

概述

本文档主要介绍 RTC 的校准方案，确保 RTC 计时准确。

说明

本文以 Hi3536 为例，未有特殊说明，Hi3521A/20DV300, Hi3531A, Hi3518EV20X/16CV200, Hi3519 V100，Hi3519V101，Hi3559V100，Hi3556V100，Hi3516CV300，Hi3531DV100，Hi3521DV100，Hi3536CV100，Hi3536DV100、Hi3520DV400、Hi3559AV100、Hi3559CV100、Hi3519AV100、Hi3556AV100、Hi3516CV500、Hi3516DV300、Hi3559V200、Hi3556V200、Hi3516EV200、Hi3516EV300、Hi3516DV200、Hi3518EV300 与 Hi3536 完全一致。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3536	V100
Hi3521A	V100
Hi3520D	V300
Hi3531A	V100
Hi3518E	V200
Hi3518E	V201
Hi3516C	V200
Hi3519	V100
Hi3516C	V300
Hi3519	V101
Hi3559	V100
Hi3556	V100
Hi3531D	V100
Hi3521D	V100

产品名称	产品版本
Hi3536C	V100
Hi3536D	V100
Hi3520D	V400
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3556A	V100
Hi3516C	V500
Hi3516D	V300
Hi3516A	V300
Hi3559	V200
Hi3556	V200
Hi3516E	V200
Hi3516E	V300
Hi3518E	V300
Hi3516D	V200

读者对象

本文档（本指南）主要适用技术支持工程师。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2019-05-15	10	2.2、2.3、5.1 小节涉及更新。 删除第 3 章和第 4 章内容。 全文增加 RTC 驱动能力调整方法。
2018-11-30	09	添加 Hi3516EV300/Hi3516EV200/Hi3518EV300 相关内容

修订日期	版本	修订说明
2018-09-26	08	添加 Hi3559V200 和 Hi3556V200 相关内容
2018-05-07	07	添加 Hi3559A/C V100 相关内容
2017-10-18	06	第 6 次正式版本发布 第 5 章节，增加说明。 增加第 6 章节。
2017-09-08	05	第 5 次正式版本发布 5.1 和 5.2 小节涉及修改
2017-08-15	04	第 4 次正式版本发布 添加 Hi3536D V100 的相关内容。
2017-07-14	03	第 3 次正式版本发布 删除 1.3 小节。 2.1 小节和 2.2 小节涉及更新。 增加第 6 章节。
2017-04-10	02	第 2 次正式版本发布 添加 Hi3536C V100 和 Hi3516AV200 的相关内容。
2016-12-01	01	第 1 次正式版本发布
2015-11-05	00B02	第 2 次临时版本发布
2015-01-19	00B01	第 1 次临时版本发布

目 录

1 概述

1.1 RTC 芯片分类

常见的 RTC 芯片，大致可分为三类：

- 非集成 RTC：只有 RTC 计时电路，不集成晶体、不集成温度补偿电路。这类芯片的计时精度主要取决于外接晶体的精度，而且受温度影响较大。通常在室温环境下，计时精度较高；随着温度升高或降低，计时偏差逐渐增大。
- 集成晶体的 RTC：将 RTC 计时电路与晶体集成，但没有温度补偿电路。这类芯片在室温环境下，计时精度更高。但仍然无法消除温度的影响。
- 集成 RTC：将 RTC 计时电路、晶体、温度补偿电路（含温度传感器）都集成在一颗芯片中，出厂时进行调教。这类 RTC 的计时精度可以做到很高，且由于温补电路的作用，受环境温度的影响很小。

1.2 RTC 工作模式

内置 RTC 可支持固定分频模式：

与非集成 RTC 相同，内置 RTC 的时钟直接采用外部晶体与振荡电路产生的经过分频后的时钟，工作时分频比固定不变。这种工作模式下，RTC 计时精度取决于外接晶体的频率精度，而且受环境温度影响。在非集成 RTC 这类芯片适用的场景，可以选择内置 RTC 替代外置非集成 RTC，节省一些器件成本。

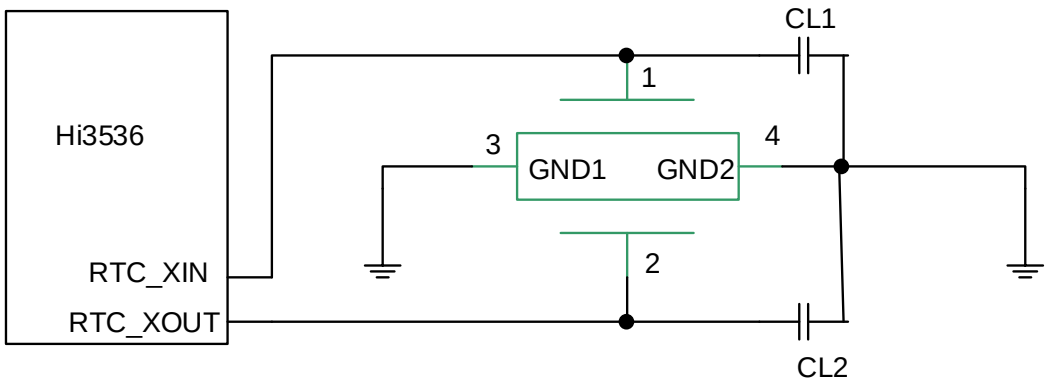
Hi3536 RTC 无内置温度补偿电路，只支持工作在固定分频模式，若 RTC 时钟的频偏较大，可通过调节 RTC 的分频系数来微调 RTC 的时钟频率。对计时精度有严格要求的客户，建议选择集成晶体的 RTC，或者带有温度补偿的 RTC。

2 RTC 的硬件参考电路

2.1 硬件参考电路

RTC 的硬件参考电路如图 2-1 所示，主要涉及晶体和电容的选择。

图2-1 RTC 晶体的硬件参考电路



2.2 选择晶体

选择晶体需要注意以下几个指标：

- 标准负载电容（Load capacitance/CL）：晶体的标准负载电容，晶体对负载电容有着严格的规定，只有实际负载电容和晶体的 SPEC 中的负载电容一致时，晶体频率才能达到标称频率。
- 串联电阻（Series resistance/Rs/ESR）：晶体的谐振腔等效串联电阻，当 ESR 越大，表示晶体越难以驱动。晶体规格中会指出 Rs 的典型值与最大值。

  芯片晶体振荡电路适用于 Rs 最大值<70KΩ 的晶体。
- 最大驱动级别（Max Drive Level/DL）：表示晶体最大的振荡幅度，当振荡幅度超过一定幅度时，晶体容易发生损坏。

晶体选型约束，如表 2-1 所示。

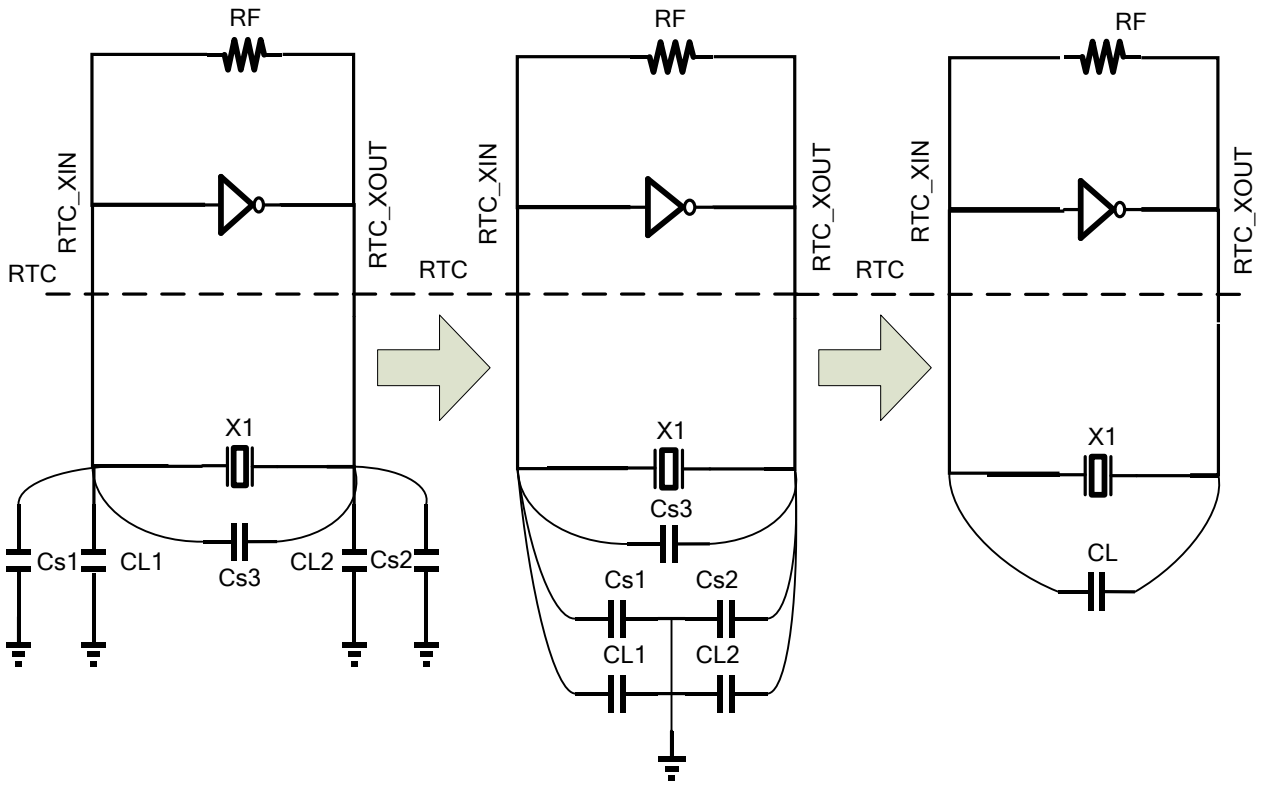
表2-1 晶体选型约束表

参数	符号	规格			
		MIN	TYP	MAX	单位
串联电阻	ESR	-	-	70	KΩ
负载电容	CL	-	12.5	-	pF
Shunt Capacitance	C0	1	-	2	pF
Motional Capacitance	C1	2	-	6	fF

2.3 选择电容

实际 CL 的示意图如图 2-2 所示。

图2-2 实际 CL 的示意图



CL_SPEC=CL1//CL2+Cs1//Cs2+Cs3

Pierce 振荡器中，一般将 CL1 与 CL2 取相同的值。

其中：

- CL_SPEC 为晶体规格书中规定的标准负载电容；
- CL1、CL2 为晶体匹配电容；
- Cs1、Cs2、Cs3 为寄生电容，一般取值为 3~5pF。

以 12.5pF 的晶体为例，CL1 与 CL2 一般取值为（12.5pF-3pF）*2=19pF。

因为寄生电容随 PCB 板设计以及晶体和主控选型不同而变化，故亦可确定 PCB 板设计以后，通过测试频偏来选取不同容值的 CL1 来获得最接近 32.768K 的输出频率，偏测试方法请参考第 5 章节“晶体相关指标测试方法”的相关内容。

3 HI_RTC 驱动使用说明

说明
Hi3536DV100/Hi3559AV100/Hi3559CV100/Hi3519AV100/Hi3556AV100/Hi3516CV500/Hi3516DV300/Hi3559V200/Hi3556V200/Hi3516EV200/Hi3516DV200/Hi3516EV300/Hi3518EV300 不支持 hi_rtc 驱动。

3.1 编译

在 RTC 目录下执行下述命令即可生成对应的驱动 hi35xx_rtc.ko、hi_rtc.ko 及示例程序 test。其中 hi35xx_rtc.ko、hi_rtc.ko 分别包括：

- hi35xx_rtc.ko：Hi3518EV20X/ Hi3516CV200、Hi3519V100、Hi3519V101、Hi3559V100、Hi3556V100 和 Hi3516CV300
- hi_rtc.ko：Hi3521A/ Hi3520DV300、Hi3531A、Hi3536、Hi3521DV100、Hi3531DV100、Hi3520DV400 和 Hi3536CV100

```
cd rtc
make
cd test
make
```

3.2 使用

将 RTC 的驱动拷贝到单板：

- 对于 Hi3518EV20X/ Hi3516CV200、Hi3519V100、Hi3519V101、Hi3559V100、Hi3556V100、Hi3516CV300，执行如下命令插入驱动模块：

  insmod hi35xx_rtc.ko
- 对于 Hi3521A、 Hi3520DV300、Hi3531A、Hi3536、Hi3521DV100、Hi3531DV100、Hi3520DV400 和 Hi3536CV100，执行如下命令插入驱动模块：

  insmod hi_rtc.ko

RTC 驱动提供的功能通过单板上运行的 test 示例程序说明，如图 3-1 所示。

图3-1 示例程序用法

```
Usage: ./test [options] [parameter1] ...
Options:
        -s(set)              Set time/alarm,      e.g '-s time 2012/7/15/13/37/59'
        -g(get)              Get time/alarm,      e.g '-g alarm'
        -w(write)            Write RTC register,  e.g '-w <reg> <val>'
        -r(ead)              Read RTC register,   e.g '-r <reg>'
        -a(alarm)            Alarm ON/OFF',       e.g '-a ON'
        -reset               RTC reset
        -b(battery monitor)  battery ON/OFF,      e.g '-b ON'
        -f(requency)         frequency precise adjustment, e.g '-f <val>'
```

设置获取时间

通过如下命令可设置 RTC 时间：

```
./test -s time <year/month/day/hour/minute/second>
```

通过如下命令可获取 RTC 时间：

```
./test -g time
```

设置获取闹钟时间

通过如下命令可设置 RTC 闹钟时间：

```
./test -s alarm <year/month/day/hour/minute/second>
```

通过如下命令可获取 RTC 闹钟时间：

```
./test -g alarm
```

通过如下命令设置闹钟到期是否产生中断，驱动中断例程由用户根据需求自由补充。

```
./test -a ON/OFF
```

读取、设置 RTC 内部寄存器

通过如下命令可读取 RTC 内部寄存器，此功能多用于辅助调试。

./test –r <reg>

通过如下命令可设置 RTC 内部寄存器，此功能多用于辅助调试。

./test –w <reg> <value>

reg 取值，请参见各芯片的用户指南的实时时钟部分。

复位 RTC 模块

通过如下命令可复位 RTC 模块。

```
./test -reset
```


固定分频模式分频系数微调设置

通过如下命令可设置分频系数从而达到调整时钟的快慢效果。

```
./test -f <val>
```

< val>值为将要设置的分频系数的 10000 倍，例如要设置分频系数为 327.60，则 val=3276000。通过直接敲“./test –f”命令可以查看当前分频系数，因为计算误差的问题，获取的分频系数可能和设置的分频系数有细小的偏差。分频系数可以配置范围为：327.60~327.70。

打开、关闭电池电量监测功能

通过如下命令可打开 RTC 电池电量监测功能。

```
./test -b ON
```

通过如下命令可关闭 RTC 电池电量监测功能。

```
./test -b OFF
```

注意

本特性仅支持：Hi3536、Hi3519V100、Hi3519V101、Hi3516AV200、Hi3516CV300 及 Hi3559V100/Hi3556V100、Hi3536CV100、Hi3531DV100、Hi3520DV400 和 Hi3521DV100

RTC 驱动能力调整方法

提供 4 档驱动能力，可以通过 SPI_RW 寄存器（各芯片地址不一样，具体请参见《Hi35xx Vxx SOC 用户指南》3.9“实时时钟”章节，以下以 Hi3516EV200 芯片地址 0X120E0004 为例）配置内部 0x1B 寄存器的低 2bit 进行调整。

步骤 1 执行下边 4 条指令，解除默认值锁定（只针对 Hi3559AV100/Hi3559CV100/Hi3519AV100/Hi3556AV100 /Hi3516EV200/Hi3516EV300/Hi3518EV300/Hi3516DV200 有效）。

```
himm 0x120E0004 0x016400CD
himm 0x120E0004 0x016500AB
himm 0x120E0004 0x0166005A
himm 0x120E0004 0x0167005A
```

步骤 2 选择需要配置的档位值，从下边的 4 条命令选择一条进行配置即可。

```
himm 0x120E0004 0x011b0000（档位0）
himm 0x120E0004 0x011b0001（档位1）
himm 0x120E0004 0x011b0002（档位2）
himm 0x120E0004 0x011b0003（档位3）
```


步骤 3　回读寄存器确认配置是否成功，下图中红框中的数值即为档位回读值。

```
himm 0x120E0004 0x019b0000
himd.l 0x120E0004
```

```
~ #
~ # himm 0x120E0004 0x019b0000
himd.l 0x120E0004*** Board tools : ver0.0.1_20121120 ***
[debug]: {source/utils/cmdshell.c:168}cmdstr:himm
0x120E0004: 0x00850000 --> 0x019B0000
[END]
~ # himd.l 0x120E0004
*** Board tools : ver0.0.1_20121120 ***
[debug]: {source/utils/cmdshell.c:168}cmdstr:himd.l
====dump memory 0x120E0004====
0000:   009b0300 009b0300 009b0300 009b0300
0010:   009b0300 009b0300 009b0300 009b0300
0020:   009b0300 009b0300 009b0300 009b0300
0030:   009b0300 009b0300 009b0300 009b0300
0040:   009b0300 009b0300 009b0300 009b0300
0050:   009b0300 009b0300 009b0300 009b0300
0060:   009b0300 009b0300 009b0300 009b0300
0070:   009b0300 009b0300 009b0300 009b0300
0080:   009b0300 009b0300 009b0300 009b0300
0090:   009b0300 009b0300 009b0300 009b0300
00a0:   009b0300 009b0300 009b0300 009b0300
00b0:   009b0300 009b0300 009b0300 009b0300
00c0:   009b0300 009b0300 009b0300 009b0300
00d0:   009b0300 009b0300 009b0300 009b0300
00e0:   009b0300 009b0300 009b0300 009b0300
00f0:   009b0300 009b0300 009b0300 009b0300
[END]
```

----结束

RTC 驱动能力代码修改方法

hi_rtc.c 文件，请配置如下寄存器，档位请配置为驱动能力调整后的档位，0x0~0x3 分别对应档位 0~3

```
static int __init rtc_init(void)
{
    int ret = 0;

    ret = misc_register(&rtc_char_driver);
    if (0 != ret)
    {
        HI_ERR("rtc device register failed!\n");
        return -EFAULT;
    }

    ret = request_irq(RTC_IRQ, &rtc_interrupt, 0,
            "RTC Alarm", NULL);
    if(0 != ret)
    {
        HI_ERR("hi35xx rtc: failed to register IRQ(%d)\n", RTC_IRQ);
        goto RTC_INIT_FAIL1;
    }

#ifdef HI_FPGA
    /* config SPI CLK */
    writel(0x1, (volatile void *)SPI_CLK_DIV);
#else
    /* clk div value = (apb_clk/spi_clk)/2-1, for asic ,
          apb clk = 62.5MHz, spi_clk = 15.625MHz,so value= 0x1 */
    writel(0x1, (volatile void *)SPI_CLK_DIV);
#endif

    /* enable total interrupt. */
    spi_rtc_write(RTC_IMSC, 0x4);
    spi_rtc_write(RTC_CLK_CFG, 0x01);

    return 0;

RTC_INIT_FAIL1:
    misc_deregister(&rtc_char_driver);
    return ret;
} « end rtc_init »
```

用户接口

请参看 hi_rtc.h 文件。

4 linux 内核标准 RTC 驱动使用说明

4.1 编译

Hi3536CV100/Hi3536DV100/Hi3559AV100/Hi3559CV100/Hi3519AV100/Hi3556AV100/Hi3516CV500/Hi3516DV300/Hi3559V200/Hi3556V200/Hi3516EV200/Hi3516EV300/Hi3518EV300/Hi3516DV200 支持内核标准 RTC 驱动，内核编译默认打开 RTC 选项。

4.2 使用

将内核烧写到单板后，启动单板，执行下面命令：

```
ls /dev/rtc0
```

能够查看到 RTC 设备，表示内核 RTC 驱动正确加载。

内核 RTC 驱动支持 ioctl 系统调用。

表4-1 内核标准 RTC 驱动 ioctl 指令功能描述

指令	功能描述
RTC_ALM_READ	读取闹钟时间
RTC_ALM_SET	读取时间与日期
RTC_SET_TIME	设置时间与日期
RTC_PIE_ON	开 RTC 全局中断
RTC_PIE_OFF	关 RTC 全局中断
RTC_AIE_ON	使能 RTC 闹钟中断
RTC_AIE_OFF	禁止 RTC 闹钟中断
RTC_UIE_ON	使能 RTC 更新中断
RTC_UIE_OFF	禁止 RTC 更新中断

指令	功能描述
RTC_IRQP_SET	设置中断的频率

设置获取时间

- 通过如下命令可设置 RTC 时间：

```
ioctl(fd, RTC_SET_TIME, &rtc_tm);
```

- 通过如下命令可获取 RTC 时间：

```
ioctl(fd, RTC_RD_TIME, &rtc_tm);
```

RTC 驱动能力调整方法

提供 4 档驱动能力，可以通过 SPI_RW 寄存器（各芯片地址不一样，具体请参见《Hi35xx Vxx SOC 用户指南》3.9“实时时钟”章节，以下以 Hi3516EV200 芯片地址 0X120E0004 为例）配置内部 0x1B 寄存器的低 2bit 进行调整。

步骤 1 执行下边 4 条指令，解除默认值锁定（只针对 Hi3559AV100/Hi3559CV100/Hi3519AV100/Hi3556AV100 /Hi3516EV200/Hi3516EV300/Hi3518EV300/Hi3516DV200 有效）。

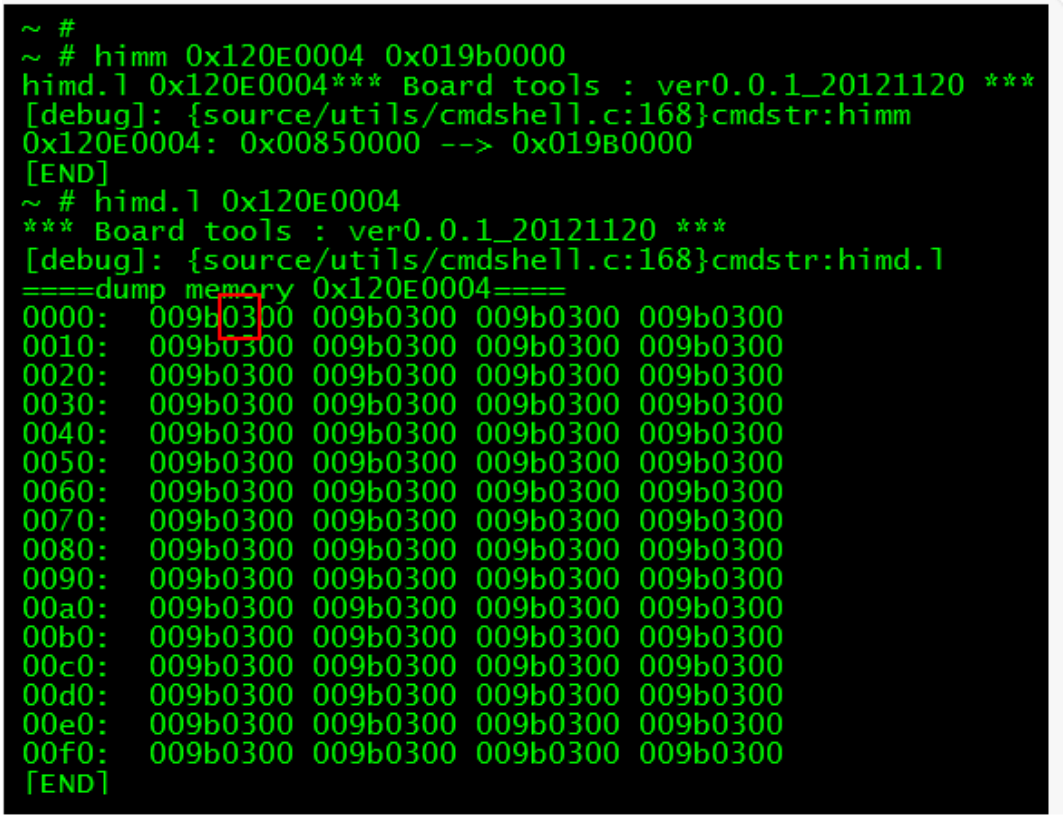
```
himm 0x120E0004 0x016400CD
himm 0x120E0004 0x016500AB
himm 0x120E0004 0x0166005A
himm 0x120E0004 0x0167005A
```

步骤 2 选择需要配置的档位值，从下边的 4 条命令选择一条进行配置即可。

```
himm 0x120E0004 0x011b0000（档位0）
himm 0x120E0004 0x011b0001（档位1）
himm 0x120E0004 0x011b0002（档位2）
himm 0x120E0004 0x011b0003（档位3）
```

步骤 3 回读寄存器确认配置是否成功，下图中红框中的数值即为档位回读值。

```
himm 0x120E0004 0x019b0000
himd.l 0x120E0004
```



----结束

RTC 驱动能力代码修改方法

\drivers\rtc\rtc-hibvt.c 文件，请配置如下寄存器，档位请配置为驱动能力调整后的档位，0x0~0x3 分别对应档位 0~3

```
static int hibvt_rtc_init(struct hibvt_rtc *rtc)
{
    void *spi_reg = rtc->regs;
    int ret = 0;
    unsigned char val = 0;
    /*
     * clk div value = (apb_clk/spi_clk)/2-1,
     *  apb clk = 100MHz, spi_clk = 10MHz,so value= 0x4
     */
    writel(CLK_DIV_DEFAULT, (spi_reg+SPI_CLK_DIV));

    ret |= hibvt_spi_rtc_write(spi_reg, RTC_IMSC, INT_MSK_DEFAULT);
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_SAR_CTRL, LV_CTL_DEFAULT);


    ret |= hibvt_spi_rtc_write(spi_reg, RTC_CLK_CFG, 0x01);

    /* default FREQ COEF */
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_FREQ_H, FREQ_H_DEFAULT);
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_FREQ_L, FREQ_L_DEFAULT);

    ret |= hibvt_spi_rtc_read(spi_reg, RTC_INT_RAW, &val);
    //ret |= hibvt_spi_rtc_read(spi_reg, RTC_CLK_CFG, &val2);
    if (ret) {
        dev_err(&rtc->rtc_dev->dev, "IO err.\n");
        return ret;
    }

    if (val & LV_INT_MASK) {
        dev_err(&rtc->rtc_dev->dev,
            "low voltage detected, date/time is not reliable.\n");
        hibvt_spi_write(rtc->regs, RTC_INT_CLR, 1);
    }

    return ret;
} « end hibvt_rtc_init »
```

用户接口

请参看内核 Documentation 目录下 rtc.txt 文件。

5 晶体相关指标测试方法

5.1 频偏测试

- 用频率计通过同轴电缆连接专门的时钟测试脚（TEST_CLK）在 uboot 下，配置寄存器，将管脚复用为 CLK_TEST 功能，并选择 32.768KHz 时钟信号输出，测试 RTC 频率。
- **强烈建议**：不要在晶振引脚上用示波器直接测量（一个典型无源探头的电容值在 10pF 之内，输入阻抗大约为 10MΩ。 两个值都会大大影响晶体振荡器的运行方式）。
- 当测得的频率过高时，负载电容的值必须增加，当测得的频率太低时，就要减少负载电容的值。客户可根据自己的产品需求来确定晶体振荡频偏范围。

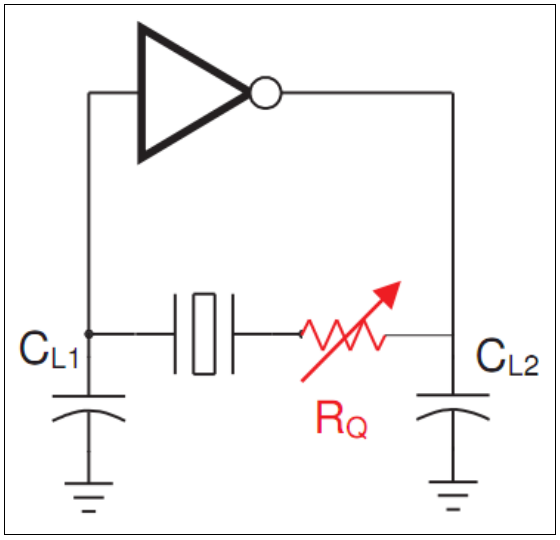
注意

在单板设计时需要将 test_clk 管脚引出测试点，方便后续晶体指标测试。

5.2 安全因子测试

增加一个与晶振串联的电阻 RQ，如图 5-1 所示。增加 RQ 的值，直到晶体恰好不起振，的 RQ 电阻最大值 RQ_{max}。（RQ_{max} 的值应该由 SMD 电阻进行验证，不建议用可调电位器）。

图5-1 安全因子测试示意图



借助于以上测试，可测得晶体振荡电路振荡裕度 (OA) 和安全因子。

振荡裕度 OA =RQ_{max} + ESR

安全因子 (SF)：

$$SF = \frac{OA}{ESR} = \frac{RQ_{max} + ESR}{ESR}$$

表5-1 安全因子约束限定条件

安全因子（SF）	限定条件
SF<2	不安全
2≤SF<3	适用
3≤SF<5	安全
SF≥5	非常安全

注意：要求安全因子至少大于 3

例：晶体 ESR 为 60K，如果串联的额外电阻 RQ_{max} 为 120K，则刚刚达到安全级别。

5.3 DL 测试

Hi3536 芯片晶体振荡电路内部限制了 RTC_XIN 与 RTC_XOUT 管脚振荡的振荡幅度，可以通过以下公式估计电路工作时的实际 Drive Level，并确定此值小于晶体规格中规定的最大 Drive Level。

$DL\_actual=0.35*Rs\_max*(\pi*f*Vpp_{XOUT}*CL)^2/2$

其中：

- Rs_max 为晶体规格书中的串联电阻的最大值。
- f 为晶体的谐振频率。
- Vpp_{XOUT} 为示波器测量的 RTC_XOUT 管脚的 peak to peak 电压。
- CL 为晶体规格书的标准负载电容。

说明

由于测试电压时示波器探头的寄生电阻和电容效应，会导致测试结果存在偏差，因此以上公式只是较简单的估算 DL 的方法，如果有需求，可以找晶体厂商进行更精确测试。

5.4 起振时间测试

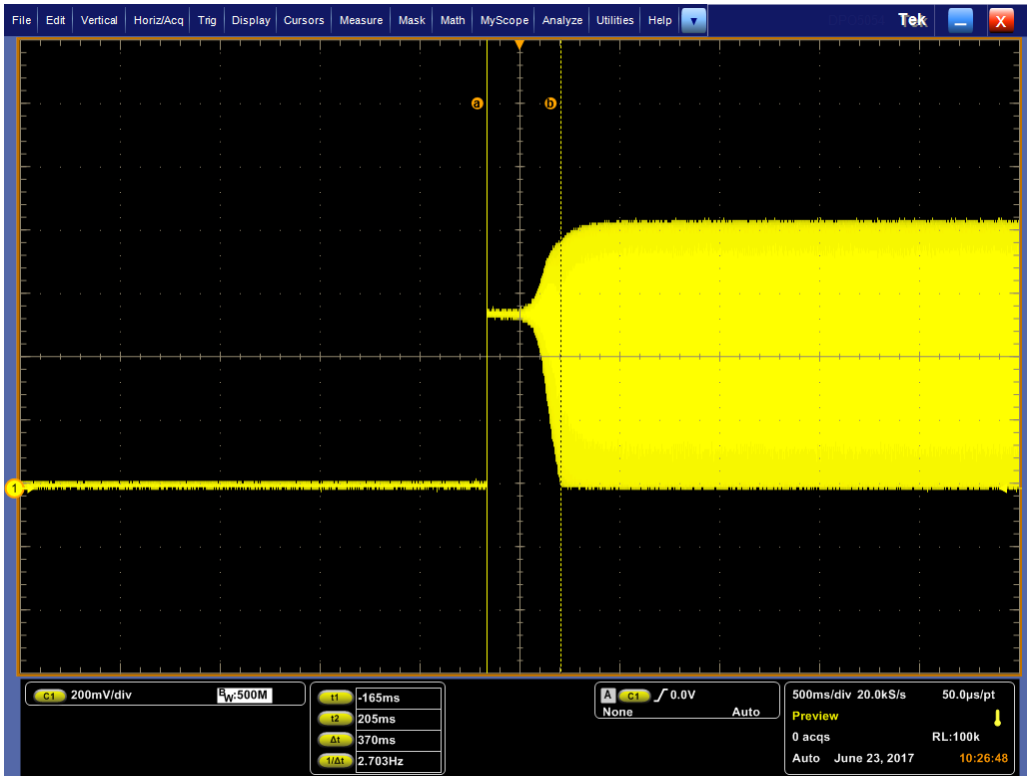
一般情况下，对于 RTC 振荡电路而言，介于几百毫秒到几秒钟之间的启动时间均为正常值，晶体振荡器的启动时间取决于不同的因素：

- 频率越低启动时间就越长（相对 24MHz 系统时钟）。
- 负载电容越大，启动时间就越长。
- 晶体 ESR 越高启动时间就越长（晶体选型时关注）。
- 振荡裕度 (OA)越大，启动快（也就是说安全因子越大，启动越快）。
- 晶体寄生电感越大，起振时间越长。

鉴于以上因素的影响，测试启动时间前，优先测试安全因子和频偏。

用示波器量测 RTC_Xout 波形，捕获波形第一个上升沿，量测从第一个上升沿到输出稳定频率的时间，如图 5-2 所示。

图5-2 起振时间示意图



6 Q&A

6.1 振荡器不振

【现象】

32.768K 时钟无输出，RTC 计时电路中的秒寄存器值恒定不变。

【分析】

使用示波器探头观察 RTC_XIN 管脚振荡波形，引起不同的振荡波形的情况有以下几种：

- 如果无振荡波形，可能是晶体损坏。
- 如果有 32K 左右频率正弦波，且 peak to peak 幅度小于 600mV，则可能是因为 CL1 与 CL2 过大，导致振荡电路驱动不足，从而幅度偏小，振荡波形无法通过后续的施密特触发器。
- 如果有 200K 左右频率的正弦波，且 peak to peak 幅度小于 600mV，则可能是因为 CL1 与 CL2 偏小，导致振荡电路振荡到 200K 频点，且由于 200K 频点振幅较小，所以无法通过后续的施密特触发器。

【解决】

- 如果确定晶体损坏，请更换晶体。
- 如果为 32K 左右频率正弦波，幅度不够，请检查 CL1 与 CL2 是否偏大，并更换正确的电容。
- 如果为 200K 左右频率正弦波，幅度不够，请检查 CL1 与 CL2 是否偏小，并更换正确的电容。

6.2 振荡器的输出频率是 200K

【现象】

32.768K 时钟输出频率接近 200K，RTC 计时电路中的秒寄存器值每秒钟增加 6。

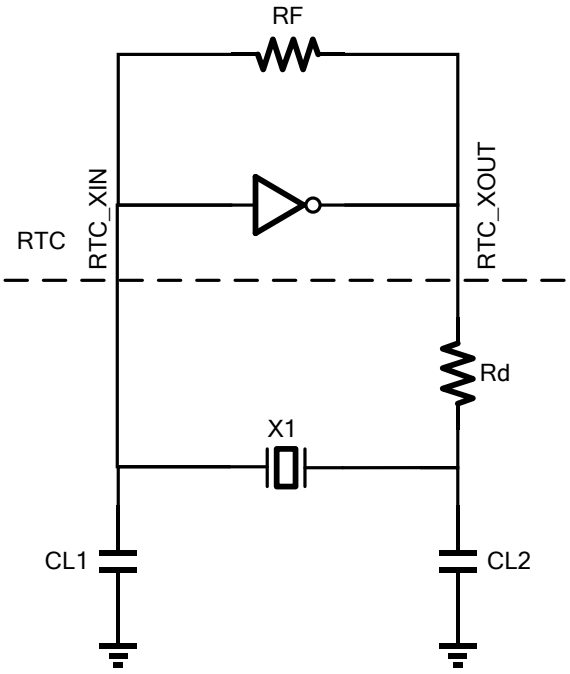
【分析】

因为 32.768K 晶体存在 6.1 倍频的谐振点，如果晶体有异常，则可能振荡到 6 倍频附近。

【解决】

建议首先检查 CL1 与 CL2 是否偏小；如果 CL1 于 CL2 为正确值，但振荡频率仍然为 200K，则可以在电路中添加如图 6-1 所示的 Rd，Rd 取值为 1/(2π x 32768 x CL2)，Rd 与 CL2 可以形成一个 RC 滤波器，降低 6.1 倍频处的环路增益。

图6-1 200K 振荡解决方案电路



注意：一般情况不建议增加 Rd，如果采用增加 Rd 的方法，需要确定 RTC_XOUT 管脚信号幅度不会过小。

6.3 振荡频率虽然是 32.768K 附近，但是频率却不准

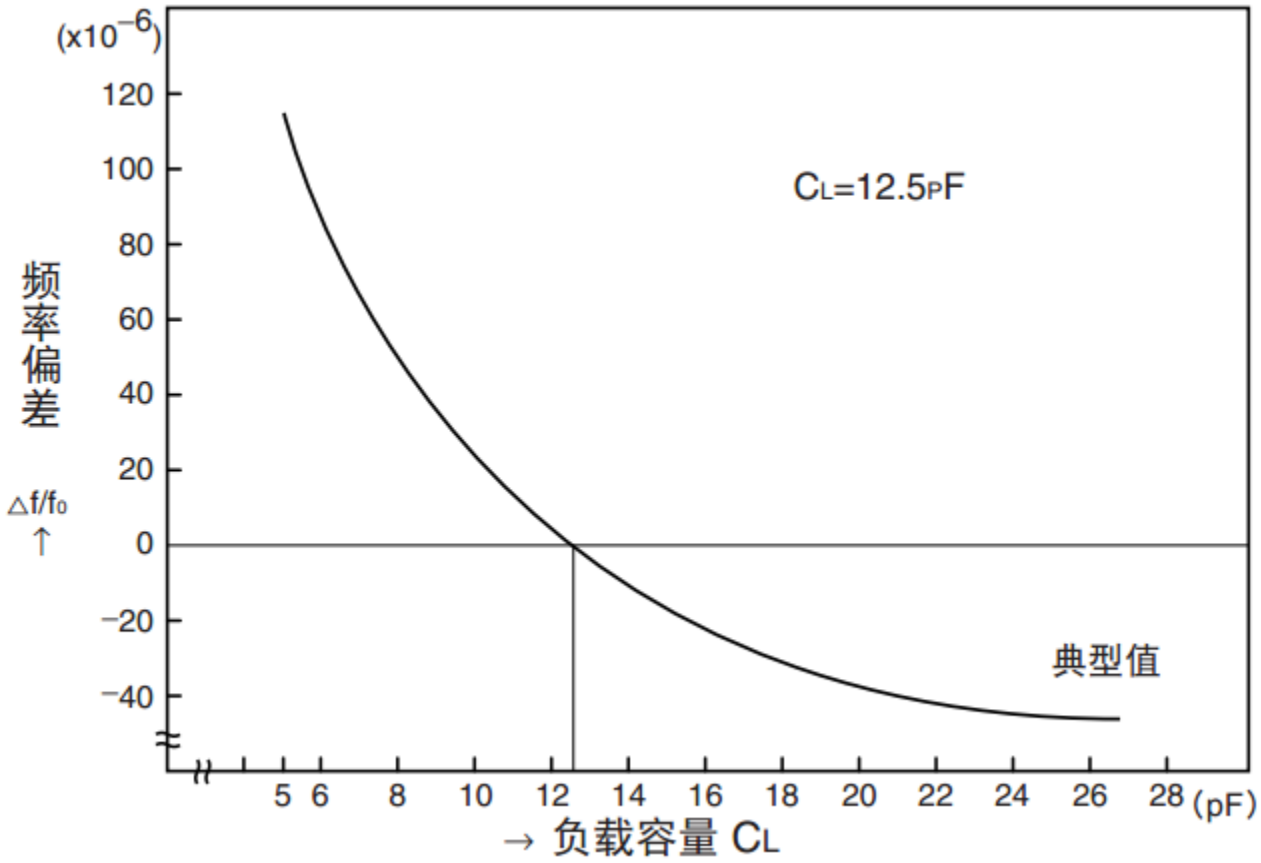
【现象】

32.768K 时钟输出频率偏离 32.768K。

【分析】

振荡电路振荡频率主要由晶体和负载电容共同保证，晶体本身确定了频率的大致范围（即图 6-2 所示中的 0 偏差对应的频率），而实际负载电容的大小则确定了频率的偏移量（即图 6-2 所示中的实际频率偏离 0 的值）。

图6-2 频率偏差和负载容量 C_L 的关系



【解决】

如果频率偏离了 32.768K，首先需要确认晶体管脚弯折不会对内部晶体部分施加应力，并确定焊接过程中的温度符合 datasheet 规范；其次，检查 CL1 与 CL2 取值是否正确，并更换正确的电容。